CLAIMS

1. A computer program product, tangibly embodied on an information carrier, comprising instructions operable to cause data processing apparatus to:

establish any number of checkpoints in a first computer program; and

include each checkpoint in a checkpoint group, wherein each checkpoint group can include any number of the checkpoints regardless of where the checkpoints are in the first computer program.

2. The product of claim 1, wherein the checkpoints comprise assertion statements and breakpoint statements.

3. The product of claim 1, further comprising instructions to:

establish activation variants to enable checkpoint groups or compilation units or both to be managed jointly.

4. The product of claim 1, further comprising instructions to:

receive a control input activating a first checkpoint group; and

activate the checkpoints in the first checkpoint group.

5. The product of claim 4, wherein the control input further specifies a mode and the mode comprises one of:

activating checkpoints that are assertions to terminate on assertion failure;

activating checkpoints that are assertions to log status on assertion failure; and

activating checkpoints that are assertions to break in a debugger on assertion failure.

6. The product of claim 4, further comprise instructions to:

receive a control input specifying a scope.

7. The product of claim 4, further comprise instructions to:

receive a control input specifying a scope specifying that activating is to be performed only for a particular user of the first computer program, that activating is to be performed only for a particular server on which the first computer program is running, or that activating is to be performed globally.

8. The product of claim 1, wherein the checkpoints and the first computer program are in a source code form.

9. The product of claim 8, wherein:

the checkpoints comprise assertion statements, each assertion statement when activated testing whether a specified assertion condition is true or false; and

the checkpoints comprise breakpoint statements, each breakpoint statement when activated halting program execution when it is encountered during program execution.

10. The product of claim 8, wherein:

the assertion statements comprise an assertion statement having an argument to activate logging with programmer-controlled granularity.

11. The product of claim 8, further comprising instructions to establish a development environment for developing the first computer program in which the checkpoint groups are development objects.

12. The product of claim 1, wherein the checkpoints and the first computer program are in a compiled form.

13. Apparatus, comprising:

means for establishing any number of checkpoints in a computer program; and

means for including each checkpoint in a checkpoint group, wherein each checkpoint group can include any number of the checkpoints regardless of where the checkpoints are in the computer program.

14. The apparatus of claim 12, wherein:

the checkpoints comprise assertions and breakpoints.

15. The apparatus of claim 12, further comprising:

means for associating an activation variant with a checkpoint group.

16. The apparatus of claim 12, further comprising:

means for associating an activation variant with a compilation unit.

17. A method, comprising:

receiving a computer program having checkpoints each identified by a group identifier,

each group identifier identifying checkpoints without limitation as to the location of the

checkpoints in the computer program, each checkpoint being an assertion or a breakpoint;

and

receiving user input to invoke checkpoints as a group according to their group

identifiers.

18. The method of claim 16, further comprising:

receiving a user input specifying a mode of invocation of checkpoints; and

invoking checkpoints according to the specified mode.

19. The method of claim 16, further comprising:

receiving a further user input specifying a scope of invocation of checkpoints, the scope

specifying that checkpoints are to be invoked only for a particular user of the first computer

program, that checkpoints are to be invoked only for a particular server on which the first

computer program is running, or that checkpoints are to be invoked globally; and

invoking checkpoints according to the specified scope.